REMARKS

The Office Action and cited references have been reviewed. Claim 1 has been amended to correct an antecedent basis. This amendment does not change the scope of the claims. Claims 1-23 remain pending and are at issue herein.

35 U.S.C. §103 Rejections

It is axiomatic that a *prima facie* case of obviousness may only be established if three basic criteria are met. First, there must be some suggestion or motivation, either in the cited references themselves or in the demonstrated knowledge generally available to one of ordinary skill in the art at the time the application was filed, to modify or combine reference teachings in the cited manner. Second, there must be a reasonable expectation of success in so doing. Finally, the prior art references, when combined or modified as asserted, must teach or suggest all the claim limitations. (*See* Manual of Patent Examining Procedure, §2143.) It is an indispensable requirement of the PTO rules, the relevant Federal statutes, and the Federal courts that the teaching or suggestion to make the claimed combination/modification and the reasonable expectation of success must be found in the prior art, not in the applicant's disclosure. (*See* In re Vaeck, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991).)

The Federal Circuit has recently confirmed that a finding of obviousness based on a combination of references must meet stringent evidentiary requirements. In particular, the Federal Circuit noted that "[t]he factual inquiry whether to combine references must be *thorough and searching*... The need for *specificity* pervades [the prevailing legal] authority." *See* In re Lee, 277 F.3d 1338, 1343, 61 U.S.P.Q.2d 1430, 1433 (Fed. Cir. 2002) (emphasis added). In the Lee decision, the Federal Circuit criticized as insufficient the examiner's conclusory statement that a combination of cited references would provide certain benefits (the same benefits that the invention provided) making the combination obvious. *See* Id. at 1434. In so doing, the Federal Circuit noted that the obviousness inquiry cannot be "resolved on subjective belief using unknown authority." Id. Rather the, PTO has an obligation and choice to either develop a solid "evidentiary basis" motivating a cited combination, or forego the rejection entirely. *See* Id. The Examiner can satisfy the burden of showing obviousness of

the combination "only by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references."

The Office Action has rejected claims 1-7, 10-11, 15-16, and 20-21 under 35 U.S.C. § 102(e) as being unpatentable over King (U.S. Patent No. 6,532,476) in view of Pettus (U.S. Patent No. 6,031,977). Reconsideration of this ground of rejection and allowance of claims 1-3, 5-8, 10-13, 15, and 17-36 in view of the foregoing amendments and the following remarks are respectfully solicited.

Regarding claim 1, the Office Action states that King '476 teaches a computer-readable medium having computer-executable instructions for performing steps to convert a stream into an N-ary tree, the stream having a list of nodes (and refers to the DynArray at col. 7, lines 63-67 of King '476), each node having a data element, a data type and a data size (and refers to col. 7, lines 63-67 of King '476), the service discovery stream having a stream size (and refers to col. 7, lines 63-67 of King '476), the steps comprising: a) retrieving the data type and the data size of one of the nodes from the service discovery stream (and refers to col. 14, lines 49-51, 59-61 of King '476); b) adding the node to a list head (and refers to col. 14, lines 61-55 of King '476);
c) performing one of decrementing the stream size by the data size and incrementing the service discovery stream to the beginning of a next data element (and refers to col. 14, lines 61-55 of King '476); and d) obtaining a next node from the list of nodes (and refers to col. 14, lines 63 – col. 15, line 3 of King '476).

The Office Action then states that King '476 does not specifically teach the stream is a service discovery stream. The Office Action states that Pettus '977 teaches a discovery stream that is converted into a tree structure at col. 11, line 64 – col. 12, line 4 of Pettus '977 and that Pettus '977 teaches that the discovery stream has data structures encoded within to represent an available service on an enabled device at col. 10, lines 19-33 of Pettus '977. The Office Action states that it would have been obvious to one of ordinary skill in the art at the time the invention was made that incorporating King '476's system for storage and retrieval of diverse information in Pettus '977's discovery service system would have expanded the directory services ability to add new services. For motivation to combine, the Office Actions states that the motivation would have been to take advantage of King's utility as an adaptive database for storing and retrieving information of any type and format.

Claim 1 requires that a service discovery stream be converted into an N-ary tree. The Examiner is directed to page 2, lines 5-13 of the instant specification where it states that a service discovery stream is a linear stream of data that has data structures encoded within it to represent an available service on an enabled device. The data structures encoded within a service discovery stream are nil (null type), unsigned integer, signed twos-complement integer, universally unique identifier (UUID), text string, Boolean, data element sequence, data element alternative, and uniform resource locater (URL). A data element sequence is a data element whose data field is a sequence of data elements. A data element alternative is a data element whose data field is a sequence of data elements from which one data element is to be selected.

The Applicants agrees that King '476 does not teach a service discovery stream. King '476 has been thoroughly reviewed. King '476 does not suggest a service discovery stream. King '476 describes problems with database storage technologies (such as "relational storage" and "Multivalue storage") and teaches a method for the storage of variable size records that are capable of storing multidimensional data to an infinite level subject to operating system constraints. King '476 teaches a non-persistent storage model that is used to structure the computer's memory to support the storage of any structure having any number of dimensions and storing any combination of types of information in a single entity. The fundamental basis of the persistent storage model of King '476 is an entity known as a DynArray (Dynamic Array), which is also referred to as a database node. The nodes contain information of items in the database. King '476 uses an example of defining city information by country, state, and county as items in the database. No teaching or suggestion of converting a service discovery stream (i.e., a linear stream of data that has data structures encoded within it to represent an available service on an enabled device) to an N-ary tree could be found in King '476.

Pettus '977 teaches a communications directory service located in each node of a network wherein the communications directory service has a tree structure to which existing directory services and other network services can be added. The tree structure has a plurality of nodes each of which has specific methods that query and browse the associated directory service if such actions are supported by the underlying service. The communications directory service also includes shared libraries which store a service object associated with each service offered on the network. Each service object of Pettus '977has a service exchange address and communication link configuration information that allows a client to use the service object to set up the communications path to the service. A client desiring to access a remote service retrieves the

appropriate service object from the communications directory service and uses the service object to set up the communications path.

Fig. 6 and col. 10, line 19 to col. 11, line 11 of Pettus '977 teach that the communications directory service module has a hierarchical directory tree 602 which allows each of the physical directory services and other services provided by a network to be located by means of a conventional tree searching techniques; a set of stack definition objects 604 which are used to program dynamically reconfigurable stacks that allow a client to request data from a remote server over a pre-specified communication link and a set of "service objects" 606. Each service object is associated with one service available on the network and contains the network address or exchange address at which the service is available and a reference 608 to one or more of the stack definition objects 604. A reference to one of these service objects is obtained by an application program desiring to access the corresponding service. The information in the object identified by this reference is then sent to the reconfigurable stack in order to set up the communication path. The directory tree 602 is organized as a single hierarchical tree such as that shown in Fig. 7 of Pettus '977.

Fig. 7 and col. 11, lines 59 – col. 12, line 4 of Pettus '977 teach that three nodes 704, 706 and 708 are shown which interact with three separate physical directory services. A fourth node, 710, is also provided, and is called a "native" namespace node. The "native" namespace node contains a reference to each of the services that are provided by the network. In order to make each service available to clients, each service is "registered" in the native namespace 710. "Registration" in the namespace as used by Pettus '977 means to insert a reference to the service into the directory where it can found by someone traversing the directory.

From the foregoing, it can clearly be seen that neither King '476 nor Pettus '977, singly or in combination, teach or suggest a service discovery stream and do not teach or suggest converting a service discovery stream into an N-ary tree as required by claim 1.

Furthermore, it can be seen from the foregoing that Pettus '977 is directed to a system that provides a single globally accessible directory service that is capable of interacting with various directory services which are provided on a network and solves the problem of the lack of a consistent globally-accessible directory of network resources that operate over heterogeneous networks without involving a user in the details of shifting from protocol to protocol as searches are performed from network to network. King '476 is directed to a system that allows records of unlimited dimensions to be constructed, maintained, and utilized in both non-persistent and

persistent storage to overcome the problems associated with "relational" storage and "Multivalue" storage. A person of ordinary skill in the art solving the problem of King '476 would have no reason to look to Pettus '977 to solve the problem. Nor would a person of ordinary skill in the art solving the problem of Pettus '977 would have no reason to look to King '476 to solve the problem. Therefore, it is respectfully submitted that there is no suggestion or motivation to combine the two references.

Therefore, in view of the above, it is respectfully submitted that neither King '476 nor Pettus '976, singly or in combination, teach or suggest all of the elements of claim 1. It is therefore respectfully requested that the Examiner withdraw the rejection of claim 1.

Claims 2-7 depend from claim 1 and are believed to be patentable for the reasons set forth above with respect to claim 1. Additionally, claim 2 further requires, *inter alia*, the step of verifying the service discovery stream. Verifying the service discovery stream requires traversing the stream linearly to make sure it is a well-formed service discovery stream. Pettus '977 at col. 9, lines 21-27 does not teach verifying a service discovery stream as the Office Action states; it teaches that the data link layer of the OSI model manipulates a raw data bit stream and transforms it into a data stream that appears free of transmission errors. A data stream that appears free of transmission errors does not teach or suggest a service discovery stream or verifying a service discovery stream. No suggestion or teaching could be found in King '476 or Pettus '977 to traverse any type of stream to make sure it is a well-formed stream.

With respect to claims 3 and 4, neither King '476 or Pettus '977 teach or suggest a service discovery stream, so neither King '476 nor Pettus '977 teach or suggest the elements of claims 3 and 4.

With respect to claims 6 and 7, the Office Action states that column 13, lines 30-35 of King '476 teach determining if the stream size of the next node is zero as required by claim 6 and determining if the data size is zero as required by claim 7. The Applicants respectfully disagree. Lines 30-35 of King '476, in conjunction with figure 20 of King '476, teach extracting a node from another node at a position y^x (position y in level x) where the leftmost position is position 1. If the node is a leaf node as that term is defined in King '476, lines 30-35 teach that a copy of the leaf is returned if the position being extracted is position 1. If the position being extracted does not exist, a null pointer is returned. The null pointer does not mean that the data size of the next node is zero; it signifies that the position to be extracted does not exist. There is no mention or suggestion of a next node in lines 30-35. Therefore, lines 30-35 of King '476 do not teach determining if the stream size of the next node is zero or if the data size is zero if the node

is not a leaf node. No suggestion or teaching could be found in King '476 or Pettus '977 to determine if the stream size of the next node is zero or if the data size is zero.

In view of the foregoing, it is respectfully requested that the Examiner withdraw the rejection of claims 2-7.

The Office Action has rejected claims 8-9, 12-14, 17-19, and 22-23 under 35 U.S.C. § 103(a) as being unpatentable over King '476 and Pettus '977, further in view of Housel, III (U.S. Patent No. 5,339,421). Reconsideration of this ground of rejection and allowance of claims 8-9, 12-14, 17-19, and 22-23 in view of the following remarks are respectfully solicited.

In the rejection, the Office Action states that the language of claims 12-14, 17-19, and 22-23 is substantially the same as previously rejected claims 1-6 and are rejected on the same rationale. It is respectfully submitted that the Examiner meant to refer to claims 8-9 in the rejection and the response to the Examiner's rejections will proceed accordingly.

Claims 8-9 depend from claim 1 and are believed to be patentable for the same reasons set forth above for claim 1. Claims 12-14 depend from claim 10 and are believed to be patentable for the same reasons set forth above for claim 10. Claims 17-19 depend from claim 15 and are believed to be patentable for the same reasons set forth above for claim 15. Claims 22-23 depend from claim 20 and are believed to be patentable for the same reasons set forth above for claim 20.

In the rejection, the Office Action has merely stated what the Examiner believes Housel teaches and did not put forth any rationale for combining references as is required to put forth a prima facie case of obviousness. Therefore, the Office Action has failed to put forth all of the requirements to make a *prima facie* case of obviousness as required by the MPEP and as pointed out in *In re Lee* and has therefore failed to put forth a *prima facie* case of obviousness. For this reason alone, it is respectfully requested that the rejection of claims 8-9, 12-14, 17-19, and 22-23 be withdrawn.

Furthermore, Housel '421 teaches a common general parser and applications programmer interface for use in data processing systems. The parser encodes data from a program for transmission onto a channel and decodes incoming data for handoff to an applications program. Column 19, line 9 to column 23, line 54 of Housel '421 teach the steps required to decode incoming data using the parse table of Figs. 13 and 14 of Housel '421. Column 20, lines 26-37 of Housel '421 specifically teaches pushing a group descriptor onto a stack. The group descriptor is used in the parser of Housel '421 that decodes incoming data for handoff to an applications program and stores the decoded data in allocated storage areas. Each group descriptor defines a

subset of other group or item descriptors and contains a group descriptor type identification that describes prescribed characteristics common to and necessary for encoding and decoding the data items identified by the item descriptors contained within the group descriptor. For group descriptors, Housel '421 teaches that the FIRST field is a pointer to the first child descriptor of the group (which may be either an item descriptor of a group descriptor). The NEXT field is a pointer to the next sibling descriptor (i.e., same hierarchal level) of the group. The FLAGS field consists of a number of control bits defined for the particular descriptor. For example, one bit indicates whether or not the group is required or optional. Other bits indicate if length or repetition checking is to be performed and if an expression should be evaluated to compute the group's length or existence.

Col. 20, lines 45-54 of Housel '421 teach that the decoder recognizes a descriptor IDT1418 is a length-type descriptor, determines if the data stream conforms to the descriptor definition by checking the type field, and derives the data length. No teaching or suggestion could be found of setting a stream size to the size of a parent node content size or a container stream size.

Col. 20, lines 62-66 of Housel '421 teach that a data stream cursor is incremented by 13, which is the number of bytes required for the length, type, and data fields and that 13 is added to an accumulator in the stack to record the total number of bytes processed. It does not teach or suggest a stream size.

From the foregoing, it can be seen that Housel '421 does not teach or suggest all of the elements of claim 8. No teaching or suggestion could be found in King '476, Pettus '977 or Housel '421, singly or in combination, to push a list head, a node, and a stream size into a stack or other elements of claim 8.

With respect to claim 9, column 21, line 41 to column 23, line 54 teach the steps of decoding templates D and J of Figs. 13 and 14 of Housel '421. Column 22, lines 20-35 of Housel '421 teach checking to confirm that the number of bytes processed for the members of a group equals the pre-determined group length and the subsequent popping of the stack. No teaching or suggestion could be found of determining if a stack is empty. Column 22, line 39 to column 23, line 8 of Housel '421 teach some of the steps of decoding the occurrence of template J. No teaching or suggestion could be found to determine if the stack is empty and obtaining a popped list head, the next node, and a popped stream size from a stack if the stack is not empty.

In view of the foregoing, it is respectfully requested that the rejection of claims 8 and 9 be withdrawn.

The Office Action states that the language of claims 10-11, 15-16, and 20-21 is substantially the same as claims 1-6 and are therefore rejected on the same rationale as the rejection of claims 1-7. The Applicants respectfully disagree.

Claims 10-11, 15-16, and 20-21 are believed to be patentable for the reasons set forth above for claims 1-7. Furthermore, independent claim 10 recites the step of adding the first node to a tail of a list if the first node is a leaf node. No teaching or suggestion could be found in King '476 of Pettus '977 to add a leaf node to a tail of a list.

Claims 11 and 16 recite the step of creating an array of nodes with the number of nodes equal to the first number of elements plus one. No teaching or suggestion could be found in King '476 or Pettus '977 to create an array of nodes with the number of nodes equal to the first number of elements plus one.

In view of the foregoing, it is respectfully requested that the Examiner withdraw the rejection of claims 10-11, 15-16, and 20-21.

The Office Action states that the language of claims 10-11, 15-16, and 20-21 is substantially the same as claims 1-6 and are therefore rejected on the same rationale as the rejection of claims 8-9. It is assumed that the Office Action mistakenly states claims 1-6 and should be claims 8-9. The Applicants respectfully disagree.

Claims 12-14, 17-19, and 22-23 are believed to be patentable for the same reasons as claims 8 and 9 and it is therefore respectfully requested that the rejection of claims 12-14, 17-19, and 22-23 be withdrawn.

Conclusion

The application is considered in good and proper form for allowance, and the Examiner is respectfully requested to pass this application to issue. If, in the opinion of the Examiner, a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,

Kevin L. Wingate

Kevin L. Wingate, Reg. No. 38662
LEYDIG, VOIT & MAYER, LTD.
6815 Weaver Road, Suite 300
Rockford, Illinois 61114-8018
(815) 963-7661 (telephone)
(815) 963-7664 (facsimile)

Date: March 3, 2005